The Annotated Ground Video Data Collection Project

David Coombs, Sandor Szabo, Martin Herman
{dcoombs, sszabo, mherman}@nist.gov
National Institute of Standards and Technology
Intelligent Systems Division
Building 220 (Metrology) Room B-124
Gaithersburg MD 20899

ABSTRACT

The goal of the Annotated Ground Video Data Collection Project, funded by Tom Strat at DARPA (Defense Advanced Research Projects Agency), is to collect and distribute video and related information that support computer vision research involving intelligent ground vehicles. The initial phase focuses on video information for image stabilization and site reconstruction. The data consist of a sequence of video collected from a side-looking camera mounted on a moving HMMWV (High Mobility Multi-purpose Wheeled Vehicle) travelling off-road and on-road at approximately 8-16 kph (5-10 mph). The camera's video is digitized and inertial navigation system (INS) data consisting of the vehicle's forward velocity and orientation are graphically overlaid in barcodes on the video signal. This annotated video is then converted back to an NTSC (National Television Standards Committee) monochrome signal and recorded on a video cassette recorder (VCR).

To produce the distributed data, the video cassette containing the annotated video is digitized at 30 frames per second (fps) at a resolution of 320x240 8-bit pixels. The digitized video sequence consists of 5708 frames of video (approximately three minutes). Videos of the data are available (see "Solar Building Data Set" and "Nike Site Data Set"). The INS data are extracted from the barcodes and stored as floating point values embedded in the last 40 pixels of each image. In addition, the camera has been calibrated using a building at the data collection site as a calibration target. The data, calibration, and supporting software are available at `http://isd.cme.nist.gov/proj/ground-video/`*. They are also available in PDF and postscript formats. (A free PDF reader can be downloaded from Adobe.)*

1 Videos

The HTML version of this document offers videos of the data that were collected in the “Solar Building Data Set” and the “Nike Site Data Set”.

The solar building video is a 40-second loop around the passive solar test building at the Nike site. Videos at 160x120 pixels (half the resolution of the recorded data) can be viewed in MPEG (0.5 MB), AVI (4 MB), and Quicktime (4 MB) formats. An MPEG video is available at the recorded resolution of 320x240 pixels (3 MB).

The Nike Site video continues around the Nike site from the solar building for 2.5 minutes. Videos at 160x120 pixels (half the resolution of the recorded data) can be viewed in MPEG (2 MB), AVI (14 MB), and Quicktime (14 MB) formats. An MPEG video is available at the recorded resolution of 320x240 pixels (10 MB).

2 Introduction

The National Institute of Standards and Technology (NIST) is assisting several government agencies in developing infrastructure and performance metrics for intelligent ground vehicles. One aspect is the development of vision based data sets that allow researchers to develop and test algorithms. In the past, most video collection efforts have consisted of simply recording video. NIST is developing systems to capture and synchronize other data with the video. Initially these data consist of the vehicle's forward velocity and orientation, collectively referred to as inertial navigation system (INS) data. The initial video and INS data collection is restricted to the currently available equipment. Experience gained in the current data collection effort is guiding the design of the next generation of data collection systems being developed at NIST. In addition, we encourage data users to provide us feedback on the data, its format, and users's anticipated future needs.

This document is organized in several sections. Section 3 describes the data collection system. Section 4 describes the Nike site where the video collection is performed. Section 5 contains details of the initial data set, including videos of the data (see "Solar Building Data Set" and "Nike Site Data Set"). Section 6 contains information on the camera calibration. Section 7 contains links to the data sets. Finally, Section 8 provides references. The remainder of this introduction discusses some of the design issues and problems encountered during the initial data collection.

Several trade-offs were considered when designing the current data collection system. We have found that data collection from our vehicle, a US Army HMMWV[1], often involves driving for several hours trying to capture the right conditions and data. Collecting video from a moving vehicle poses significant technical challenges. The main challenges are the storage and bandwidth requirements for storing large amounts of video. These are compounded by the fact that the system must work on a vehicle driving under rugged conditions that can prove hazardous to typical mass storage devices.

In designing a collection system, one might first consider recording the video on tape and capturing the INS data in memory. The primary drawbacks of this approach are the inability to synchronize the INS data with the video and the limited amount of memory available for INS data. Some VCRs advertise time stamp capabilities but the systems we examined did not allow computers to read time stamps and there were doubts the computer could get time stamp information at frame rates.

[1] Certain commercial equipment, instruments, or materials are identified in this paper in order to adequately specify the experimental procedure. Such identification does not imply recommendation or endorsement by NIST, nor does it imply that the materials or equipment identified are necessarily best for the purpose.

We chose to overlay a time stamp and data on the video signal. The time stamp is a graphic binary barcode which is incremented each frame. In theory, the time stamp can be easily extracted from a video signal using simple vision processing algorithms. In practice, it proved difficult. When we digitize the video, we have trouble synchronizing to the video signal. This results in captured frames often consisting of the odd field of one frame and the even field of another. Since the time stamp is changed every frame, the captured barcode is interlaced, with odd lines belonging to one frame and even lines belonging to another. We verified that the codes were cleanly written to the tape, we used the highest quality S-VHS tape available, we used different VCRs, and we employed a time-base corrector (TBC). In the end, we decided to initially offer half-resolution video images because the data are consistent within any single video field. The next generation video and motion data collection system should support full-resolution video.

The graphically overlaid time stamp ensures synchronization of video and INS data. The trade-off is the loss in video signal quality due to the overlay procedure we employ. We digitize the camera signal and then reconstruct a monochrome NTSC signal prior to recording on video tape. Digital video technology has advanced considerably and we expect in the near future to capture video and other data and store them in a completely digital format.

3 Data Collection System

The current data collection system consists of a side-looking video camera, a Datacube MV-200 image processor, an inertial navigation system (INS) called MAPS (Modular Azimuth Positioning System), a Motorola MV162 processor board, and a S-VHS video cassette recorder (VCR). The procedure for data collection is to overlay INS data, in the form of a graphic binary barcode overlay, on each video frame. The reason for overlaying the data on the video, rather than storing the data in a separate file, is to simplify archiving and correlating data with video. The data are always available with the video (even if the video is on tape) and the data are always synchronized with the video. The drawbacks are the loss of image field-of-view occupied by the over lay and the signal degradation due to conversions between analog and digital formats. We are currently developing a digital data collection system in which digital data will be stored in only one or two lines of each frame without degradation from conversions. Details of the current collection system are presented here.

3.1 Camera

The camera is an Elmo MN401E "lipstick" camera. It is mounted left-looking atop the cupola (cab) above the driver's seat. An 8.5 mm C-mount lens (with adapter) provides a field of view about 40 degrees wide. The camera controls are set to full auto-white-balance, AGC (automatic gain control) is off, and the shutter speed is 1/100 s. To cope with the low angle of the winter sun, a camera visor was mounted just above the camera field of view to reduce "starbursts" in the video data. The aperture is nearly closed.

3.2 INS

The inertial navigation system (INS) is a Modular Azimuth Positioning System (MAPS) that is commonly used to position and aim Howitzers. The MAPS contains three ring laser gyros, three accelerometers and a rear axle odometer. The MAPS can supply orientation and translation data at 5 Hz rates with a translation resolution of one meter. Orientation data alone are available at 25 Hz. To obtain a faster update rate and a higher resolution, Alliant Tech Systems developed a Navigation Interface Unit (NIU). The NIU requests only the orientation data from the MAPS and reads translation data from the odometer. The NIU integrates the odometry and orientation data, providing position and heading at 25 Hz ([4]).

The following information is available from the NIU:

```
struct
```

```
{
 double d;                    /* total distance traveled, m */
 double v;                    /* current velocity, m/sec */

 double x;                    /* north, vehicle position, m */
 double y;                    /* east, vehicle position, m */
 double z;                    /* down, vehicle position, m */

 double th;                   /* vehicle heading, radians */
 double p;                    /* vehicle pitch, radians */
 double r;                    /* vehicle roll, radians */

 double xd;                   /* current component, m/sec */
 double yd;                   /* current component, m/sec */
 double zd;                   /* current component, m/sec */

 double xdd;                  /* current component, m/sec/sec */
 double ydd;                  /* current component, m/sec/sec */
 double zdd;                  /* current component, m/sec/sec */

 double fx;                   /* front of vehicle */
 double fy;                   /* front of vehicle */
 double fz;                   /* front of vehicle */
}
```

The currently collected data are velocity, heading, roll, and pitch. This subset was initially chosen to support camera stabilization. Any of the data in the struct can be collected.

3.3 Collecting Video and INS data

The video and INS data are collected by a system comprised by a Motorola MV-162 host processor running VxWorks and a Datacube MV-200 pipelined image processor mounted in a VME chassis. The host downloads a program to the MV-200. This program configures the MV-200 to digitize video as it arrives from the camera, to move individual frames into display memory, to combine overlay memory with display memory, and to convert the combined digital display into analog NTSC (on the green channel). Once the MV-200 is running, the host is interrupted each time a frame is digitized. The host reads the most recent INS data from the NIU, converts the requested data (presently velocity and orientation) into a binary barcode graphic overlay and writes the overlay into the MV-200's overlay memory.

During the data collection, the gain of the MV-200 analog-to-digital converter can be adjusted. For the collected data, the gain was set to 1.0.

3.4 Video Cassette Recorder

A Panasonic AG-5700 portable VCR is used to record the data in S-VHS format. The video is recorded on Sony broadcast quality tape.

3.5 Video Digitization and Data Extraction

The video and INS data currently maintained on the web site are in digital form. These data are digitized at half resolution on a Sun video card (320x240). Digitizing the sequences and saving them to disk exceeds the bandwidth of our Sun Ultra Sparc. This limitation is overcome by creating a program that digitizes frames, extracts frame counts and detects when frames are dropped. The program then automatically rewinds the tape until the lost frame is found. Disk space storage and video tape capacity are the only limits on the length of a video sequence that can be digitized.

The files are in Sun raster format. Each video frame consists of a raster header, the image data and the INS data. (In the initial half-resolution data, each frame consists of only a single NTSC video field.) The INS data are also stored in the last 40 pixels of each frame. This format is easily viewed with `xvq.` (Derived from `xv, xvq` is short for `xv-quick`.) The format of the data files, sample programs to read the files and plot data, and `xvq` are described in Section 5.

4 Data Collection Site

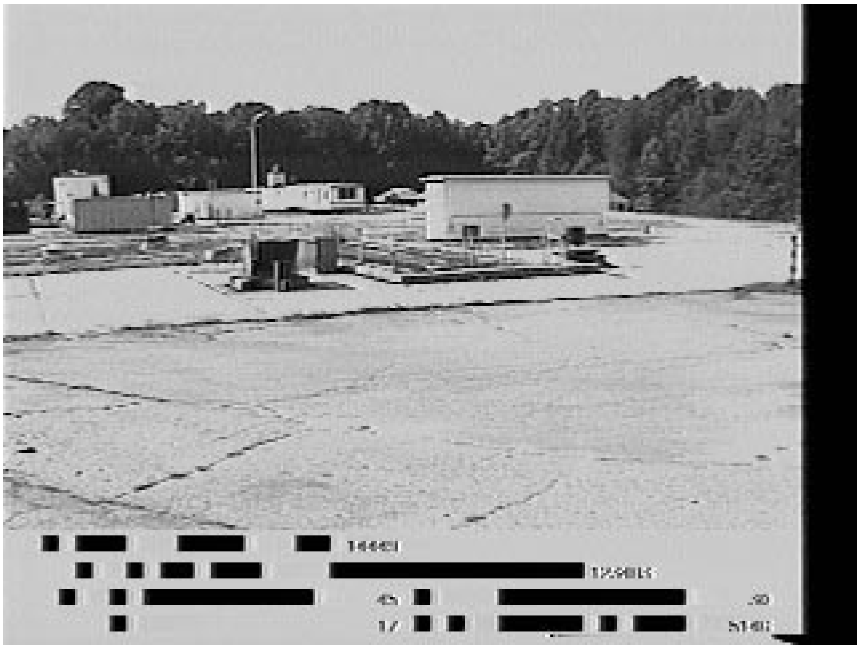

Figure 1View of the NIST Nike site, which has a handful of small buildings scattered across the asphalt pad and the surrounding grass.

The data are collected at NIST’s Nike site (shown in Figure 1). The site has a raised level asphalt pad of roughly 100 by 200 meters. On this pad are some permanent buildings and some trailers. The pad is surrounded by a strip of turf tens of meters wide on which are located several permanent structures. The passive solar test building, off one corner of the pad, has been measured to calibrate the camera (Section 6).

5 Initial Data Set

Two data sets are currently available (see Section 7). Each data set consists of a sequence of video and INS data combined in a single file. The format of these files and a brief description of their contents follow.

5.1 Data File Format

Each data file consists of a sequence of records, each record containing a raster image header, a raster image, and INS data captured at the time the image was digitized.

A Sun raster header describes the format of the raster image. Including the header with each image is redundant but it does simplify moving images between files and applications. Below is a brief description of information contained in the header.

```
/* raster_hdr[0]           /* magic # == 1504078485      */
/* raster_hdr[1]           /* width                      */
/* raster_hdr[2]           /* height                     */
/* raster_hdr[3]           /* bits per pixel             */
/* raster_hdr[4]           /* length of image in bytes   */
/* raster_hdr[5]           /* RT_STANDARD raster type    */
/* raster_hdr[6]           /* RMT_NONE    colormap       */
/* raster_hdr[7]                                         */
```

The size of each raster image is 320 columns by 240 rows. Each pixel is a byte representing 8 bits of grey scale.

The INS data consists of the following:

```
typedef struct {
 int frameCnt;                 /* counter incremented each frame */
 int bbkTimeStamp;            /* 5 ms time stamp */
 double v;                    /* current velocity, m/sec */
 double th;                   /* vehicle heading, radians */
 double p;                    /* vehicle pitch, radians */
 double r;                    /* vehicle roll, radians */
} ANNOTATED_DATA;
```

Each `int` is four bytes and each double is eight bytes for a total of 40 bytes per INS data set. These are embedded in the last 40 pixels of each raster image.

5.2 Source code

Several included C programs illustrate reading the data files and interpreting the INS data. The programs are available with the data in a compressed `tar` file. (See Section 7.) The following is a brief description of the code.

5.2.1 dataUtils.c

`dataUtils.c` contains routines to read video and INS data from files.

5.2.2 getFileSubSet.c

`getFileSubSet` extracts a subset of a data file. For example, the first 1250 frames of the larger Nike site data file were extracted to make the smaller solar building data file. The program is an example of using routines in `dataUtils.c`.

5.2.3 graphData.c

`graphData` creates INS data files suitable for graphing with `xgraph`. Also included is `grafit`, a `/bin/csh` script that parses multi-column data files, converts them to single column data files suitable for `xgraph,` and executes `xgraph`. `graphData` shows how the heading, roll, and pitch INS data can be used to plot a path of the vehicle's trajectory (see Figure 2 and Figure 3).

5.2.4 xvq.c

`xvq` is a front end to `xv`, an X Windows image tool. `xvq` can directly read the data files. The INS data embedded at the end of each image can be seen as small pixel changes at the bottom of the `xvq` window.

5.3 Solar Building Data Set

The solar building data set, illustrated in Figure 2, was extracted from the Nike site data set (see Section 5.4 below). It consists of the first 1250 frames (40 seconds, 96 MB) of the Nike site data, which comprise the initial loop around the solar building. This data set was extracted to simplify testing software and evaluating the INS data. Videos of these data at 160x120 pixels (half the resolution of the recorded data) can be viewed in MPEG (0.5 MB), AVI (4 MB), and Quicktime (4 MB) formats. An MPEG of the video data is available at the recorded resolution of 320x240 pixels (3 MB).

5.4 Nike Site Data Set

The Nike Site data set, shown in Figure 3, begins with a loop around the solar building and goes on to loop around a portion of the Nike site perimeter. The coordinates of the path in the figure were extracted from the data files. Since the data contain only forward velocity and orientation of the vehicle, velocity was integrated to obtain position. The path estimation takes into account vehicle pitch

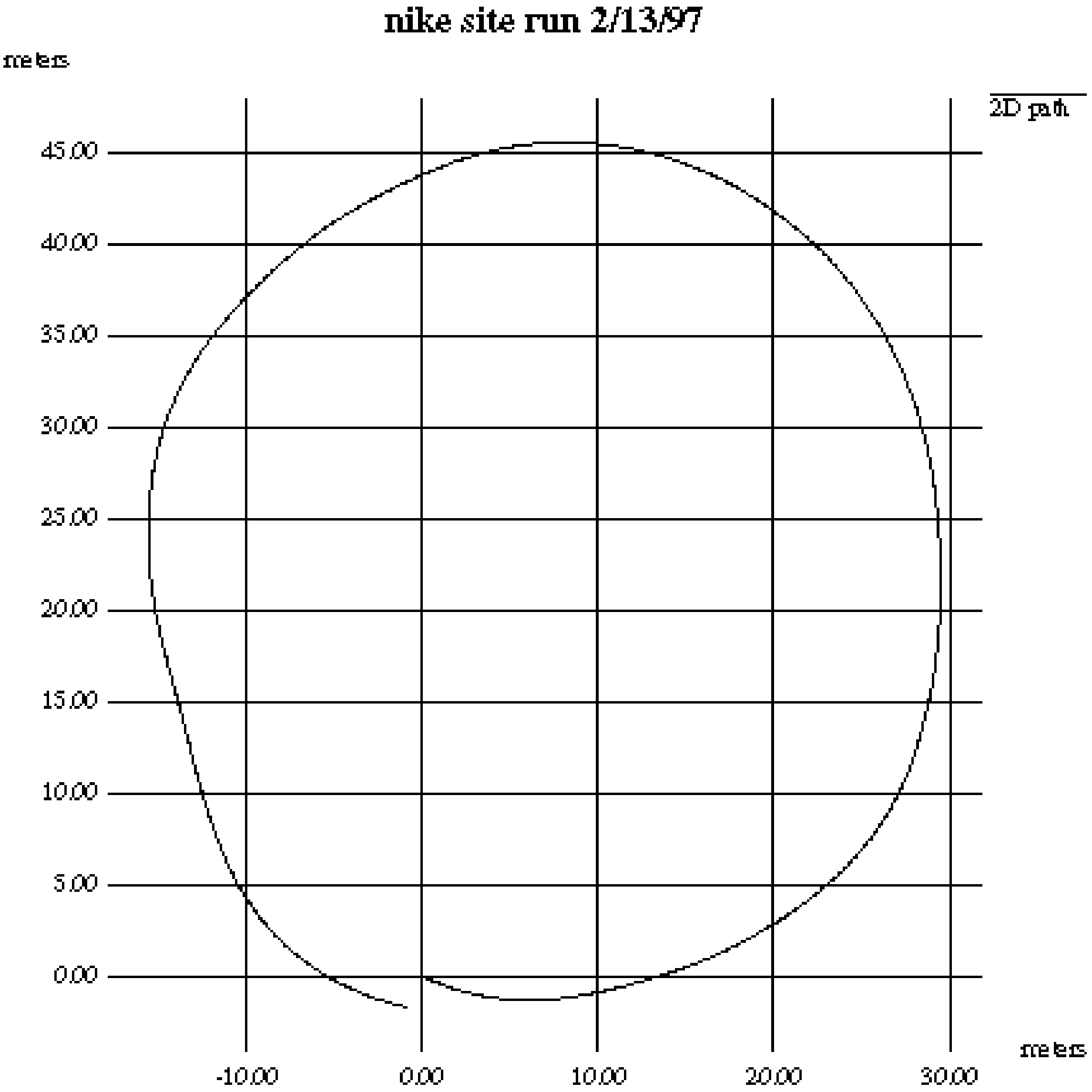

Figure 2 Solar building data set consists of video and INS data recorded during one loop around the Nike site solar building. This data set is the first 1250 frames of the longer Nike site data set.

in computing the vehicle’s *x,y* position. The file contains 5708 frames (three minutes, 440 MB) of video and INS data. Videos of these data at 160x120 pixels (half the resolution of the recorded data) can be viewed in MPEG (2 MB), AVI (14 MB), and Quicktime (14 MB) formats. An MPEG of the video data is available at the recorded resolution of 320x240 pixels (10 MB).

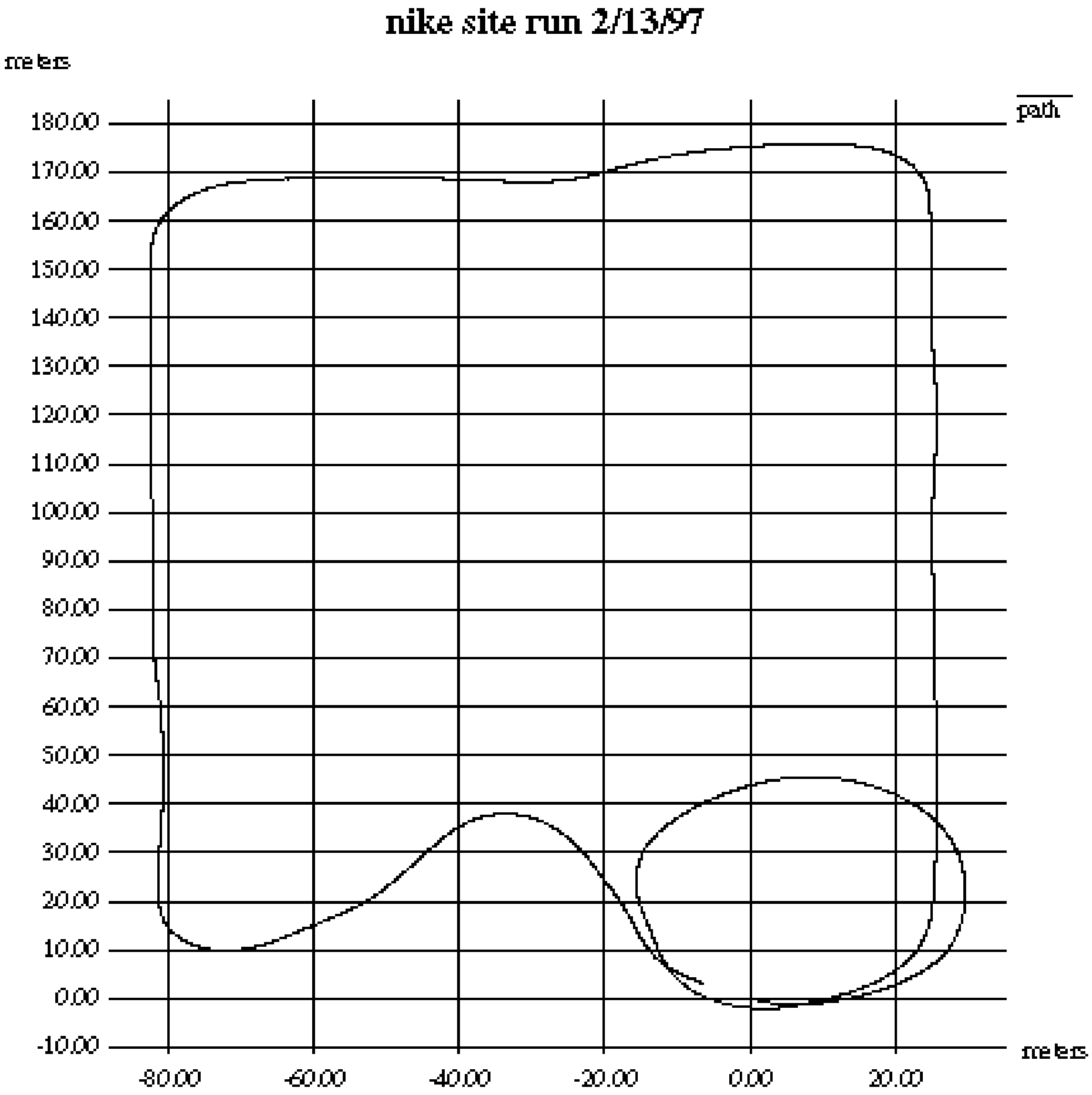

Figure 3 The path driven by the vehicle when video and INS data were collected. This path was derived from INS data extracted from an annotated video tape

6 Camera Calibration

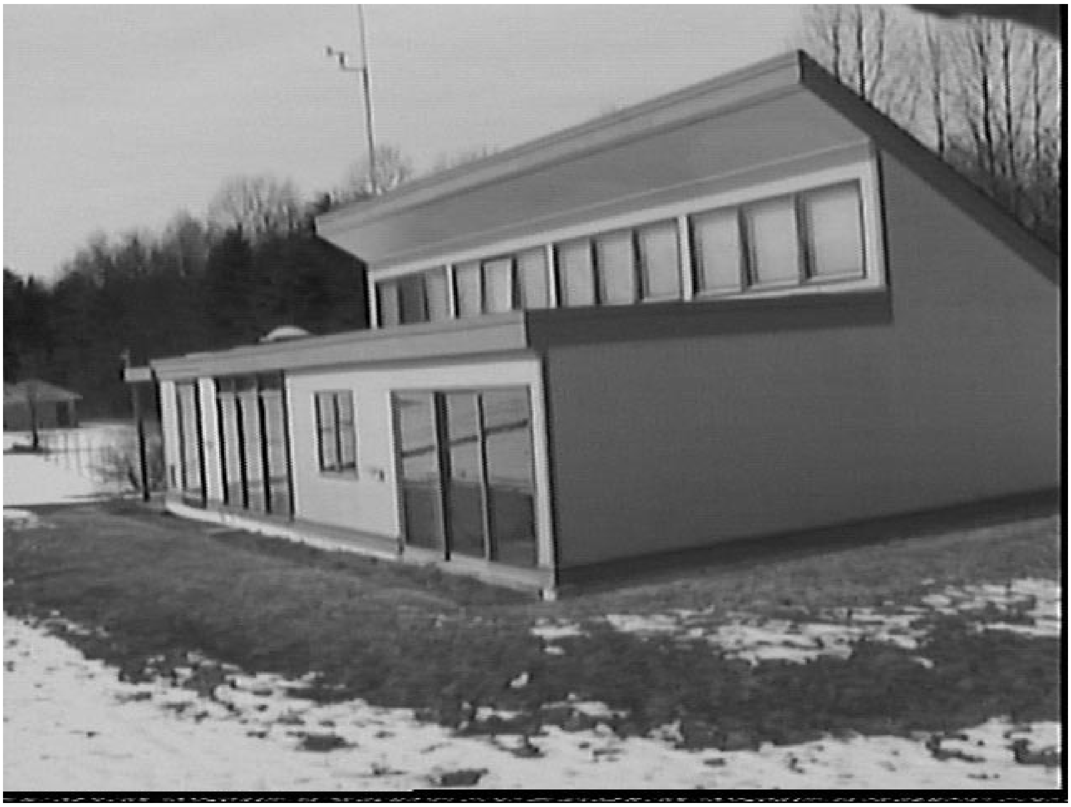

Figure 4 Calibration target: measurements were taken manually on the passive solar test building at the Nike site. The features consist mostly of corners of windows, doors, and trim on the building. In this view, 64 features are visible on the front face, the second story face, and the end of the structure. The structure is about 20 meters wide and 9 meters deep.

The camera was calibrated using Roger Tsai's method[1][2] as maintained by Reg Willson[3]. See the Tsai Camera Calibration FAQ for detailed guidance in using this method. The 3D world coordinates were manually tape-measured on the solar test building shown in Figure 4. The world coordinate system (CS) is a right-handed CS with *x* positive rightward, *y* positive upward, and *z* positive forward (toward the building's front face from its rear face). The front face has the windows and doors. The image *U* and *V* axes are coincident with the camera *X* and *Y* axes with the origin in the upper left corner of the image, with *U* positive rightward, and *V* positive downward. The 2D image coordinates of the corresponding points were manually estimated using `xv` to view the image and read the cursor coordinates. All the data, code, and results are available at `http-files/`. The calibration procedure is outlined below.

6.1 Calibration Procedure

Select a calibration target at the site. Select a building with sufficient distinctive corner features. These features will need to be relatively simple to pick out by either automatic or manual means. Select a viewing angle that offers depth range of data points on the same order as the camera standoff from the calibration object in order to aid estimating the radial distortion. Attempt to fill the camera field of view. The solar test building has been chosen for this site. See the Tsai Camera Calibration FAQ for further guidance in selecting a calibration target and the camera viewpoint.

Tape measure the coordinates of distinctive corner features of the calibration target. Manually note dimensions on previously captured printed images of the target in the units of the tape measures (feet and inches).

Capture images of the calibration target on video tape at the start of the data collection run.

Roll the video of the calibration target and digitize an image from tape using `SV` to store an image in raster format. Convert the image to any desired formats with `xv` or another image tool. (*NB:* the commands shown here are meant to illustrate the procedures. The `pwd` is `http-files/nike97feb13/calib/`.)

```
SV -s1 -f1 -I0 > calib.rs
```

View the calibration image with `xv` or `photoshop` to find image coordinates *(U, V)* of feature points and manually record them.

```
xv calib.rs &
```

Select an arbitrary world coordinate system in which to record the 3D coordinates of the calibration target. (See guidelines in the Tsai Camera Calibration FAQ for locating the world CS origin.) The world CS is a right-handed CS with origin at the left rear corner of the solar test building, with *x* positive rightward, *y* positive upward, and *z* positive forward (toward the building's front face from its rear face). This conveniently produces positive coordinates for most features of the building. (The origin is at the height of the building slab, at the bottom of the base trim. The porch pad is 15.24 cm (6 in) lower. The origin is located at a virtual corner formed by the back face of the building and the left face of the entry airlock.) Manually enter the feature coordinates in a text file as quintuples of floating point numbers *(x, y, z, U, V)*.

```
emacs calib-ncc-cd-ft.dat
```

Verify that the data are approximately correct to catch transcription errors. Bear in mind the low precision of manually choosing image coordinates and tape-measuring 3D coordinates of features. If you don’t have `Mathematica 3.0` to read the Mathematica notebook, you can download `MathReader 3.0` (available for Macintosh and Windows 95/NT platforms as of 11 Aug 1997).

```
mathematica verify-data.nb
```

Convert the units from feet to mm with `conv_cd_ft2mm` (which we added to the calibration code).

```
conv_cd_ft2mm calib-ncc-cd-ft.dat > calib-ncc-cd-mm-df.dat
```

Perform calibration with `nccal` and `nccal_fo`. Calibrating with both methods permits comparison of the results. Capture both the data output (`stdout`) and text output (`stderr`). Make sure to edit the code of each calibration program to use the correct camera and frame grabber parameters. We have added a function to set the parameters correctly for the Elmo and Sunvideo.

```
/bin/sh
nccal calib-ncc-cd-mm-df.dat > calib-ncc-cpcc-df.dat 2> calib-ncc-
cpcc-df.txt
nccal_fo calib-ncc-cd-mm-df.dat > calib-ncc-cpcc-df-fo.dat 2> cal-
ib-ncc-cpcc-df-fo.txt
exit
```

It is possible that the default origin is located too near the image center. (See guidelines in the Tsai Camera Calibration FAQ for locating the world CS origin.) To check this, `conv_cd_ft2mm` also translates the world CS origin to the right front, the right rear, or the right center of the data. It is necessary to edit the `makefile` to define the macro that will select the desired origin location and recompile `conv_cd_ft2mm`. Then simply repeat the conversion and calibration procedures.

For example, to use the right front lower corner of the data as the origin, define `TRANS_RF` in the `makefile` and proceed.

```
pushd ../../camera-calib/Tsai-method-v3.0b3/
touch conv_cd_ft2mm.c
make conv_cd_ft2mm
popd
conv_cd_ft2mm calib-ncc-cd-ft.dat > calib-ncc-cd-mm-rf.dat
/bin/sh
nccal calib-ncc-cd-mm-rf.dat > calib-ncc-cpcc-rf.dat 2> calib-ncc-
cpcc-rf.txt
nccal_fo calib-ncc-cd-mm-rf.dat > calib-ncc-cpcc-rf-fo.dat 2> cal-
ib-ncc-cpcc-rf-fo.txt
exit
```

To place the origin at the right rear lower corner of the data, define `TRANS_RR` in the `makefile` and proceed:

```
pushd ../../camera-calib/Tsai-method-v3.0b3/
touch conv_cd_ft2mm.c
make conv_cd_ft2mm
popd
conv_cd_ft2mm calib-ncc-cd-ft.dat > calib-ncc-cd-mm-rr.dat
/bin/sh
nccal calib-ncc-cd-mm-rr.dat > calib-ncc-cpcc-rr.dat 2> calib-ncc-cpcc-rr.txt
nccal_fo calib-ncc-cd-mm-rr.dat > calib-ncc-cpcc-rr-fo.dat 2> calib-ncc-cpcc-rr-fo.txt
exit
```

To place the origin at the center of the right lower edge of the data, define `TRANS_RC` in the `makefile` and proceed:

```
pushd ../../camera-calib/Tsai-method-v3.0b3/
touch conv_cd_ft2mm.c
make conv_cd_ft2mm
popd
conv_cd_ft2mm calib-ncc-cd-ft.dat > calib-ncc-cd-mm-rc.dat
/bin/sh
nccal calib-ncc-cd-mm-rc.dat > calib-ncc-cpcc-rc.dat 2> calib-ncc-cpcc-rc.txt
nccal_fo calib-ncc-cd-mm-rc.dat > calib-ncc-cpcc-rc-fo.dat 2> calib-ncc-cpcc-rc-fo.txt
exit
```

7 Initial Data Distribution

Data, code, and documentation are available at `http://isd.cme.nist.gov/proj/ground-video/`. In addition, two `tar` files are available at `ftp://isd-ftp.cme.nist.gov/pub/ground-video/`. The larger file is a compressed tar of the entire ground-video tree including both the Nike site perimeter data and the solar building data prefix of the perimeter data. When uncompressed, this entire tree is about 600 MB. The smaller file has everything except the large perimeter data set, so it is only about 170 MB when uncompressed.

8 References

[1] Roger Y. Tsai, “An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision”, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Miami Beach, FL, 1986, pages 364-374.

[2] Roger Y. Tsai, “A versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses”, *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 4, August 1987, pages 323-344.

[3] R. Willson. Public domain implementation of Tsai camera calibration. `http://www.cs.cmu.edu/~rgw/TsaiCode.html`

[4] K.N. Murphy, “Navigation and Retro-Traverse on a Remotely Operated Vehicle”, *IEEE Singapore International Conference on Intelligent Control and Instrumentation*, February 1992.